```
**************************************************************
*                                                            *
*          QQQQQQQQ        ZZZZZZZZZ      XX      XX       *
*         QQQQQQQQQQ       ZZZZZZZZ       XX    XX         *
*         QQ      QQ             ZZ        XX XX           *
*         QQ      QQ            ZZ          XXX            *
*         QQ      QQ           ZZ           XXX            *
*         QQ    QQ QQ        ZZ             XX XX          *
*         QQQQQQQQQQ        ZZZZZZZZ       XX   XX        *
*          QQQQQQQQ         ZZZZZZZZZ      XX     XX      *
*                QQ                                          *
*                                                            *
**************************************************************
```

The Journal Covering Amateur Radio & Sinclair Computers
ZX-80; Micro-Ace; ZX81; T/S 1000, 1500, & 2068

2025 O'Donnell Drive
Las Cruces, New Mexico
88001

K5XY
Alex. F. Burr
Publisher

W4GHV/5
Bo Barry
Editor

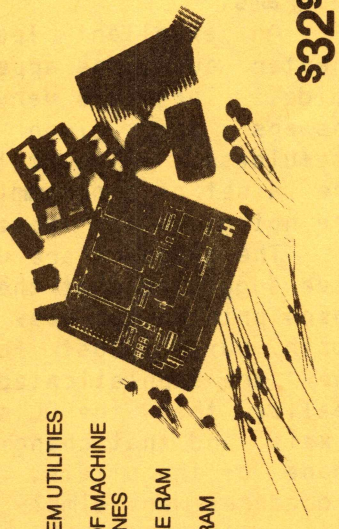
Volume 3 Number 1 ($1.50/issue) December 1984

* The QL Instruction Book, K5XY
* Editor's Doings, W4GHV
* QZX Friends
* Unclassified Ads
* Complete RTTY for the Timex, KA1VW (reprinted from QEX)

Extend your ZX81/TS1000/TS1500: Add Memory that won't Forget!

➧ *DESCRIBED IN* **Radio-Electronics** *JULY/AUGUST 1983*

- ✓ ADD YOUR OWN SYSTEM UTILITIES
- ✓ BUILD UP A LIBRARY OF MACHINE LANGUAGE SUBROUTINES
- ✓ UP TO 8K NONVOLATILE RAM
- ✓ USE HM6116LP CMOS RAM OR 2716/2732 EPROM
- ✓ COMPATIBLE WITH 16K RAM PACKS



$32.95
plus $1.95 shipping and handling

➧ *READ THE REVIEWS:*

What a super product!... conceived and executed very nicely...and with quality components.
(SYNTAX QUARTERLY Winter 82)

8K Nonvolatile memory is a gem! It has so many possible uses...I recommend this board most heartily.
(OKLAHOMA S.U.G. Newsletter 1/3)

We found the documentation to be far superior to that (of) most hardware we've received.
(S.U.N. Newsletter Nov/Dec 82)

For versatility this is even better than an EPROM...ranks quite high on the list of "must-haves".
(SYNC Magazine Mar/Apr 83)

Provides the user with instant software...an extremely versatile memory extension...
(Z-WEST June 83)

INTRODUCTION

This memory board is designed to fill the transparent 8K block of memory (from 8K to 16K) in a ZX81-16K system. This area of memory is an ideal place to store, either permanently or temporarily, machine language routines or data which are to be used by the BASIC system.

Sample utilities are included with the kit.

The use of HM6116LP 2K CMOS RAM memory IC's with their own reserve power supply means that routines stored in the RAM are nonvolatile — the RAM retains its memory even when the ZX81 is switched off or reset. Moreover, being RAM, the routines you store in the memory are easily modified. The lithium cell supplied with the board will maintain sufficient reserve power for almost ten years.

ASSEMBLY

Complete step-by-step instructions in a 20 page manual make assembly of the board easy. The kit (pictured above) is complete with a silkscreened solder-masked printed circuit board; all capacitors, resistors, transistors, sockets, connectors, integrated circuits; and the lithium cell. The board is supplied with one 2K CMOS 6116LP-3 RAM — it will accomodate three more for a total of 8K.

Complete kit with one 2K 6116LP-3	$32.95
Additional three 6116LP-3	~~$69.00~~ $0.00
Bare pc board & manual	$13.05
Kit for EPROM use only	$22.95
Assembled & tested with 2K	$47.95
Assembled & tested with 8K	$85.95
Shipping & handling per order	$ 1.95

Send check or money order to the address below:

HUNTER, 1630 FOREST HILLS DRIVE, OKEMOS, MI 48864

THE QL INSTRUCTION BOOK

Alex. F. Burr K5XY

We have obtained a Xerox copy of the QL Instruction book. It is really a very good book; although it is a bit light on the technical details which our readers like. However, for the first time computer user, it has a well written introduction to all aspects of the QL. For the user already familiar with computers, a quicker path into the details of this new system is provided.

The book came in a substantial three ring binder printed on good weight paper. The contents were divided into nine parts separated by tabbed dividers. It is possible that the final version of the instructions released in the United States will be in a different form, but I expect the contents to be unchanged.

The sections were headed

1. Beginner's Guide
2. Keywords
3. QL Word Processing
4. QL Spreadsheet
5. QL Data Base
6. QL Graphics
7. QL Concepts
8. QL Add On's
9. QL Info

In addition there was a section called QL Introduction, but no index. There is not even a good table of contents.

The contents of most of these sections will be unsurprising to anybody who has read the extensive reviews given here in previous issues. The reassuring fact is that all the features which were described here have been implemented in some fashion.

The introduction is just that. It deals with plugging in the power supply (a line cord transformer with no on/off switch, connection to a TV set, and a list of items in the box. These items include three little plastic feet which can be inserted into the bottom rear of the QL to tilt the keyboard at a fashionable angle. These are a nice touch. Similar simple items for such computers as the TRS 80 Model 100 sell for $3.50 a set.

The beginners guide covers such things as the use of the return key, reset switch (yes the QL has one), and the organization of the screen on the TV set or monitor. This section is the largest because it is here that the user is taught how to create SuperBASIC programs. SuperBASIC as the name of the programming language is properly written, is definitely an extension of the BASIC language we are all familiar with. I will comment on new features which this language possesses next month. Suffice it for now to say that there are some major changes in programming.

Keywords section takes all the SuperBASIC command words and defines them. This is a reference section. Programming techniques were covered in the previous chapter. However many clarifying explanations are given along with the definitions.

The next four sections discuss in detail the use of Quill (the word processing program, Abacus (the spreadsheet), Archive (the data base manager), and Easel (the graphics program). The features of these programs have been covered in previous articles. Easel appears to be even better than its write ups. The interchange of data between these programs is perhaps a bit more difficult than was hoped for.

QL Concepts is a short reference chapter on some features not discussed elsewhere and contains some of the more technical details of other concepts. Such things as the RS 232 interface, the character set, clock, color (or colour as the British say), graphics, and the memory map are covered. Input, output, and ROM cartridges are discussed. Even the operating system is covered (but in only two pages).

The short Add On's chapter gives more input/output details and mentions that Sinclair is preparing to give more details to vendors and serious home users.

The QL Information chapter is mainly a pitch to join the official users group Qlub, the guarantee, and some more details of information exchange between software programs.

An excellent feature is the end of chapter questions appearing in the Beginner's Guide. These are very good study aids. The answers are given to increase their usefulness. Also at the end of most chapters are a set of programming problems. Answers are not given for these.

The graphics are very good. Some people have suggested that the computer (with an FX80 Epson printer for easy screen dumps) is almost worth buying just for the graphics alone. They are sophisticated. It is important to realize that the QL screen has non square pixels and that changing mode will change the shape of the pixels. Thus if the graphics procedures were simply pixel based, they would draw different shapes in the two modes. For example, in one mode we would have a circle while the same figure in the other mode would be an ellipse.

The graphics procedures insure that whatever screen mode is in use, consistent figures are produced. It is not possible to use a simple pixel count to indicate sizes of figures. Instead the graphics procedures use an arbitary scale and coordinate system to specify sizes and positions of figures.

While the graphics capabilities of the QL are much greater than the T/S 2068, the sound capabilities are not. The technical sound capabilities are described in less than two pages. Only one frequency at a time can be produced. The lack of prominence given to sound is indicated by the fact that sound is controlled by the command BEEP. BEEP is versatile, however. The output is controlled by specifying:

A pitch - up to two pitches.

The ramp - the rate at which the sound must move between the pitches.

The wrap - how the sound is to behave after it has reached one of the specified pitches.

Randomness - deviation from the ramp.

Fuzz - deviation on the cycles established by the wrap.

Even Sinclair does not know what sound can be produced. They say "QL sound is best explored through experiment. Slight changes in the value of a single parameter can have alarming results on the sound generated, don't panic!"

All is not perfect. Nothing ever is. There were a rather disappointingly large number of typographical errors, many of which prevent some of the simple sample programs from running. Experienced users expect such things and can spot them. Inexperienced users will just become even more frustrated.

Several people have commented that the micro cartridge can be write protected in a manner similar to an audio cassette. Unfortunately if one tries to write to a write protected cartridge, the QL will continue to try without reporting anything wrong until you remove the cartridge.

I noted that pin 6 of the RS 232 interface has +12 volts on it. This pin is labeled by the standard as DATA SET READY. I hope that the source of that 12 volts will withstand a short and that it will not put out enough current to damage true RS 232 devices put across it.

In all, I, as as many others, are eagerly awaiting the actual introduction of the real QL.

Editor's Doings

Alex & I would like to wish all of our subscribers a Merry Christmas and a prosperous New Year. We are looking forward to 1985, planning to continue our 'faithful service' to QZX subscribers round the world. By the way, those of you that have contributed to QZX will be hearing from us soon. Thanks a bunch for all the excellent material you have sent!

To insure your receipt of our 'Seasons Greetings' we 'fudged a bit'. Our trusty print shop, Tresco Inc., here in Las Cruces, who has bent over backwards to help us, will be closing down for the holidays soon and I promised to get the December issue to them in time for them to get it out before they closed. And some of you may agree that your trusty editor & publisher need a little rest too!

So, we are reprinting an excellent RTTY article that appeared in the September 1984 issue of the ARRL publication **QEX**. Those of you that already subscribe to it or are not interested in RTTY will of course find **QZX** fast reading this month. (You deserve the rest too.) For those not familiar with **QEX** we have included the subscription form, you may want to give it a try. I always enjoy reading it and think most of you will do the same.

Have a Merry Christmas, and squelch the gossip that **QZX** is always a month late! With a little luck the January issue will be received in January! 73, Bo - W4GHV

QZX FRIENDS

K7HSX -- Cliff says that he really liked the Oscar 10 Plot program. It was quite impressive.

JA1ANG -- Haruo has been using his ZX 81 for about two years and it has become indispensable when he needs orbital predictions of all OSCAR satellites and the RS series of satellites. The software being used is the AMS-81 (AMSAT) and GM4IHJ's programs for several different satellites. He says that you can find him whenever you have a common window on a downlink frequency of approximately 145.940 MHz.

VE3EG -- Ray renewed early so he wouldn't have to be bothered to do it during his winter vacation in Florida. He even included a little extra to help cover the extra postage which we have to pay for Canadian subscribers. Thanks.

K0OV -- Joseph says that he would like to use his ZX81 in the mobile with a portable TV and a DF triangulation program for T-Hunting.

N6ASB -- Phil is an old QZX friend who saw our new address in Worldradio News and rejoined. He is looking for T/S 2068 RTTY info.

W6IAN -- William is another new subscriber with a T/S 2068.

KB4XH -- John is using his T/S 2068 with a Gemini-10X printer via a homebrew interface. He also has a cartridge Spectrum emulator.

W6OOA -- A. L. has a T/S 1000 which he uses on 40 m CW with a KC4UG interface and program. He operates from a mobile home.

KA2LHO -- Kraig says that his recovery from his accident is progressing nicely even though he had to have a second operation on

his leg and he still has to walk with a cane. He asked us to forward his thanks to all the QZX subscribers who sent him get well wishes. He also says that he recommends QZX on the info sheets he sends out on his HAM HELPER series of T/S programs. (Best wishes Kraig, hope you get well soon, 73, Bo)

KA8SKV -- Carl is a new member who likes to build and is retired from a successfull printed curcuit business. He has generously offered to design and build cucuit boards for QZX member projects at cost. He has a T/S 1000. He says he has been in radio for 60 years and really enjoys construction and technical projects.

KA6PKL -- Bill has a ZX81 and thinks the CWSS program works super.

UNCLASSIFIED ADS

WANTED: UNCLASSIFIED ADS for publication in QZX. Cost is 1 cent for each keystroke (characters, blanks, etc.). Payment must be included with order. This is an excellent way to sell your program or unwanted equipment, ask for help, or find a specific program. Send your ad with payment to QZX, 2025 O'Donnell, Las Cruces, NM 88001.

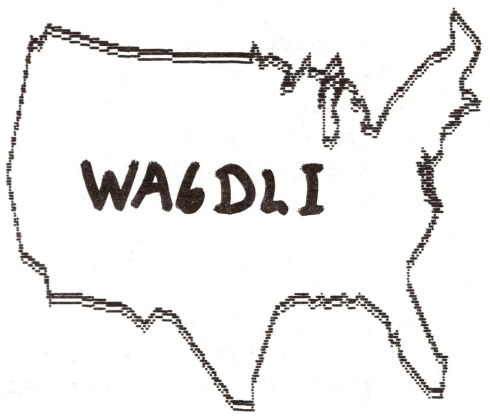
FOR SALE: ZX81/TS1000 edgeboard wire-wrap connectors; keyed; bifurcated contacts; manufactured by TI; $5.00 each. Pc piggy-back extender boards; $1.50 each. Economical experimenting with Radio-Shack's 276-158 and 161 project boards and these connectors. Add $0.75 shipping per order. Paul Hunter, 1630 Forest Hills Drive, Okemos, MI 48864.

FOR SALE: CWSS users: MLI speeds up your program and gives you a delete function. Don't XMIT your mistakes, DELETE them! MLI cassette and simple instructions- $7.50 ppd. John McGinley / 203 Milwaukee St / Mauston, WI 53948.

FOR SALE: TS1500 + 16 K Ram + TS2040 3 mos old. All or separately $60, $15, $60 or $125.00. Ron Forbis NW4M, 2418 Carmel Rd., Birmingham, AL 35235.

FOR SALE: KA2LHO HAM HELPER SERIES-13 ham radio programs from Novice Exams to electronics to County Hunting to antenna design, All written in easy to modify BASIC for the ZX-81, TS 1000, TS 1500, and TS 2068. This is one of the few complete lines of ham software available for the 2068. For complete listing including 4 additional non-ham programs and misc. other hardware/software items, send a business size SASE to KA2LHO HAM HELPER SERIES, Kraig D. Pritts, 3421 Oneida St., Chadwicks, NY 13319.

INFO WANTED: Information needed on interfacing TX 1000 with wind speed indicator and battery voltage information. Bill Scott, KA6PKL, 947 Martin St., Manteca, CA 95336.




```
    1 CLS
    2 LET N=0
    4 LET G=9
    5 REM U.S. MAP
    8 PAPER 1: BORDER 1: INK 7: C
LS
    9 LET A=INT (RND*30)+20
   10 PLOT 46,76-A/2: DRAW -2,10:
 DRAW -10,6: DRAW -6,28: DRAW 10
,46: DRAW 3,-2
   11 LET N=N+1
   16 IF N=3 THEN LET G=10
   20 DRAW 4,6: DRAW 31,-8: DRAW
29,-2: DRAW 26,0: DRAW 1,4: DRAW
 2,-2: DRAW 1,-4
   30 DRAW 5,2: DRAW 8,-4: DRAW -
4,-7: DRAW 4,1: DRAW 1,2: DRAW 7
,8: DRAW 0,-5: DRAW 6,-1: DRAW 6
,6
   40 DRAW 2,-6: DRAW -9,-4: DRAW
 -1,-6: DRAW 2,2: DRAW 0,-14: DR
AW 3,-2: DRAW 3,4: DRAW 0,4: DRA
W -2,4: DRAW 1,6: DRAW 1,-1: DRA
W 2,5
   50 DRAW 4,-2: DRAW 2,-3: DRAW
-2,-3: DRAW 2,-1: DRAW 2,2: DRAW
 3,-6: DRAW -3,-3: DRAW 6,-2: DR
AW 6,10: DRAW 16,14: DRAW 4,2
   60 DRAW 4,1: DRAW 3,3: DRAW -1
,8: DRAW 1,3: DRAW 11,-11: DRAW
-2,-5: DRAW -6,-8: DRAW 1,-4: DR
AW 3,0: DRAW 4,-4
   70 DRAW -4,0: DRAW -8,-4: DRAW
 4,0: DRAW -4,-4: DRAW -2,0: DRA
W -2,-4: DRAW 0,-4: DRAW -4,-4:
DRAW -4,4: DRAW 4,-8: DRAW -2,-4
   80 PLOT 206,116-A/2: DRAW 0,4:
 DRAW -4,2: DRAW 2,-2: DRAW 0,-4
: DRAW 3,0: DRAW 2,-4: DRAW -4,0
: DRAW 5,-2: DRAW 2,-2
   90 DRAW -4,0: DRAW 2,-2: DRAW
-4,-8: DRAW -8,-10: DRAW 0,-7: D
RAW 12,-24: DRAW 4,-2: DRAW -2,-
11: DRAW -5,0: DRAW -12,12: DRAW
 -2,8
  100 DRAW -2,-2: DRAW -2,7: DRAW
 -12,8: DRAW -2,-2: DRAW -2,0: D
RAW -2,2: DRAW -8,-2: DRAW -2,2:
 DRAW -4,-2
  110 DRAW -2,2: DRAW -5,0: DRAW
-8,-4: DRAW -10,-25: DRAW 0,-8:
DRAW -8,0: DRAW -25,30: DRAW -8,
2: DRAW -5,-5
  120 DRAW -5,4: DRAW -5,3: DRAW
-4,0: DRAW -15,3: DRAW -10,0
  130 GO TO G
```

Complete RTTY for the TIMEX

By Thomas R. Strohl,* KA1VW

This article presents the hardware and software that I have developed to use with my Timex TS-1000 computer. The software runs on a 2k byte unexpanded system, and listings are available for both Baudot and ASCII operation from the ARRL. [1]

Functions of the System

The hardware interface is based on the Z80A SIO serial input/output device. The SIO contains a dual channel USART, with control registers, handshake control lines, and data buffers for each channel. The device has features that make it quite suitable for amateur use, but its full capabilities are far beyond the necessities of amateur RTTY. In asynchronous operation it automatically adds start, stop, and parity bits (if parity is enabled). It is programmable with respect to stop bits per character, parity, bits per character, and data clock rates (1, 16, 32, or 64 times the data baud rate). Thus, the device can be programmed for ASCII as well as Baudot with any combination of stop and parity bits in use by amateurs. It also allows additional protection against spurious signals generated by faster data clock rates.

Since the Z80A SIO is part of the Z80 family, it is easy to use with the Z80A CPU -- the heart of the TS-1000. Many of the pins of the SIO connect directly to the CPU bus. All inputs and outputs are TTL compatible with the necessary pull-up resistors self contained. It requires a single 5-V dc regulated power supply that can be derived from the TS-1000 9-V supply. In the circuit presented here, the address decoding is simple and straight forward.

There are actually four versions of the Z80A SIO designated /Ø, /1, /2, and /9. Internally they are similar, but the main difference among them is that the /9 is a single channel version. Each channel has 3 read registers and 7 write registers. The read registers provide information about the data passing through the device such as buffer overflow, framing errors, and parity errors. In the asynchronous mode, read register Ø is polled to determine when a received character is available when receiving or if the transmit buffer is empty when transmitting. Only write registers 3, 4 and 5 need be programmed for asynchronous operation in the polling mode. Complete information about the Z80A SIO can be found in the "Z80 SIO Technical Manual." [2]

The price of the SIO depends on the supplier. Prices range from $12.50 to $25 for the /Ø version used here. It pays to search for the best price and I have included pin functions for all the versions should you want to make a substitute. My unit was built at a cost of $25 using breadboard construction and the lowest-priced SIO.

Review of the Circuits

Fig. 1 shows the minimum circuit required for RTTY operation. It will interface any TTL compatible terminal unit to the TS-1000, and permits operation at one speed only, in excess of 300 baud. Fig. 2 shows modifications for multiple baud rates and software TR switching.

Address decoding is performed by U1, U2, and U3. This scheme was adapted from that of J. J. Carr. [3] When address bits Ø, 1, and 6 are high and bits 2, 4, 5, and 7 are low, then all eight inputs to U1 will be high. This causes the output of U1 to go low.

If this should happen when an input/output instruction is being performed by the CPU, the input/output request (IORQ) line also goes low. The two low inputs to the NOR gate (U3) cause its output to go high. This output is inverted and serves as both the chip enable (CE) and IORQ for the SIO. Note that bit A3 can be either high or low. The result is that a single channel of the SIO has two port addresses, 67 and 75. Port 67 is used to pass character data between the computer and the terminal unit. Port 75 is used to pass programming information to the SIO and for passing buffer information to the computer.

As may be apparent, the wiring of the address lines and U2 determine the interface addresses. If you wish to use port addresses other than 67 and 75, be sure the ones selected are not being used by the Sinclair Logic chip. The system will crash if you try to write to an address that is already in use. If you try to read from such an address, a value other than 255 will be returned.

All data lines, M1, RD, ground, and the CPU clock are connnected directly to the CPU bus. Power is supplied from a 7805 5-V regulator connected to the 9-V pin of the bus. The edge connector I used was made by cutting a larger connector purchased from Radio Shack (0.1 inch spacing). A prototyping board, marked edge connector, and expansion board (necessary if you want to attach a printer or RAM pack) are available. [4] If you cut your own, it is a good idea to fit it with a key for the slot to be sure the connections line up properly.

The data clock is built using a 555 timer IC. The clock pulses do not need to be perfectly

*Rural Route 1, Box 123, Hope Valley, RI 02832

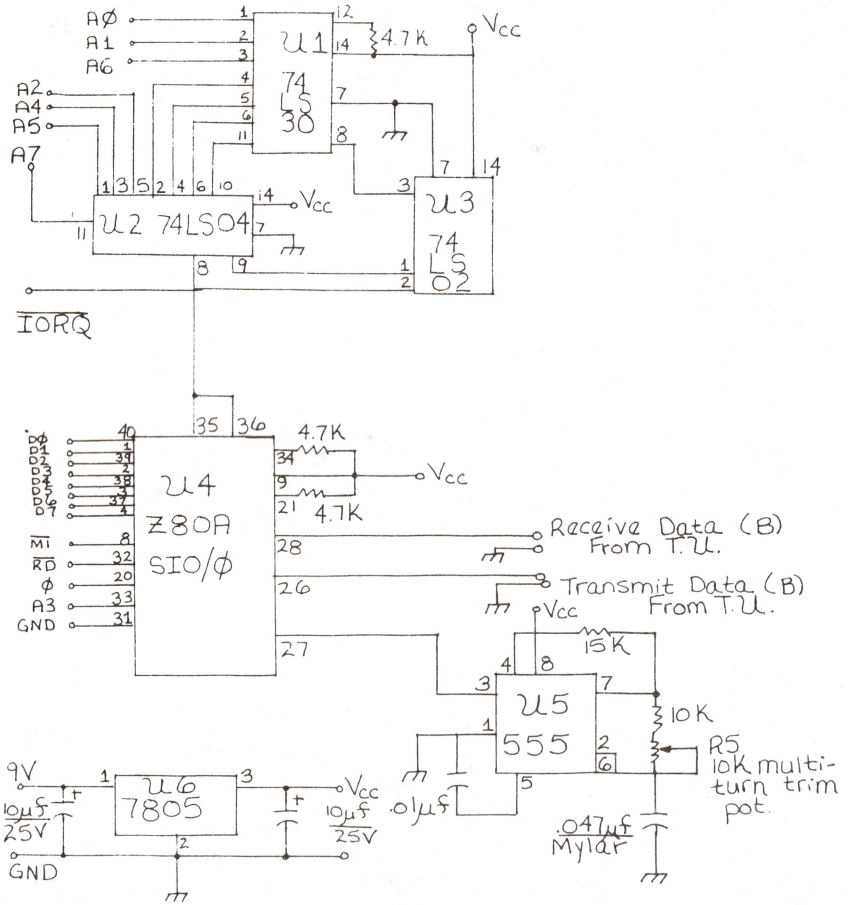
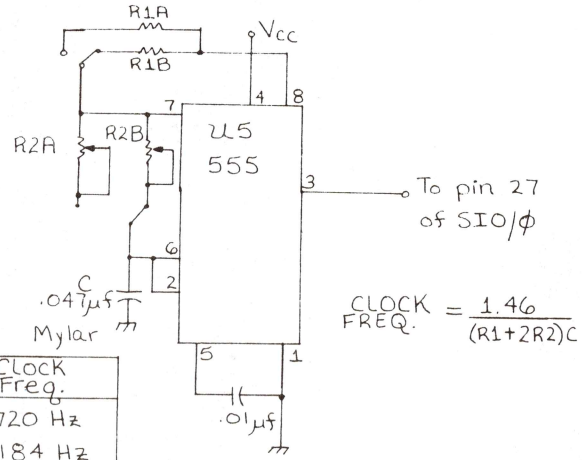


Fig. 1 — Timer circuit values are for 45-baud (60 WPM) operation.



R1	R2 (multi-turn)	Baud Rate	Clock Freq.
15K	20K	45	720 Hz
15K	10K	74	1184 Hz
6.2K	10K	110	1760 Hz
2.2K	5K	300	4800 Hz

Fig. 2A -- The clock circuit for selecting two different baud rates. This circuit may be expanded to provide for any number of clock frequencies.

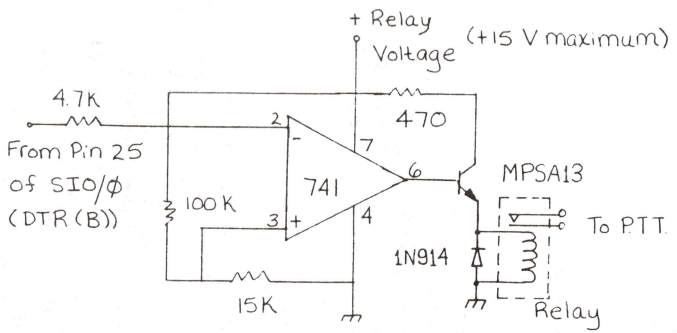


Fig. 2B — Circuit to provide software TR switching. Software required is listed on page 9. The 470- resistor may need to be changed to ensure relay activation.

symmetric and after two years of use I have not found any frequency drift in the one I am using. The clock rate should be adjusted with R5 to run at 16 times the data rate (720 Hz for 60 WPM Baudot).

The Baudot Program

Space requirements will not allow an explanation of how the machine language programs work. I present only the details of how to use them, some customizing that can be done, and the program listings themselves. If you are interested in how it works, I have added comments to the listing to help you understand it. Let's first look at how the Baudot program is operated once it's on tape.

The program is loaded using the name "Baudot." After loading, it starts to run automatically in the receive mode. The upper ten lines of the display show the received text, printed on line ten and scrolled automatically. The software searches for and scrolls on a received space after 21 characters have been printed on the line. This almost completely eliminates half words at the end of the line. If no space is received, scrolling takes place after printing 32 characters on the line. Line 11 is blank and separates the received text from the transmit buffer text.

The next ten lines display the contents of the transmit buffer. As you type on the keyboard, the typed text appears across the bottom line and scrolls up as the line is filled. When the buffer is full, text is not accepted from the keyboard or from the string load routine. Once nine full lines have been typed, the buffer cannot be written to until the whole top line has been sent. As each character is transmitted, its appearance in the buffer is changed from black to white to white on black. This allows you to keep track of what is being sent and how much room is available in the buffer.

To change from receive to transmit, press shift-S (as if to type the keyword LPRINT). You may continue to type into the buffer, space allowing, while characters are being sent. Pressing shift-D (SLOW) will cause transmission of the buffer to cease and return to the receive mode. When the transmit mode is again selected, transmission from the buffer will resume where it left off. There is no way to delete characters from the buffer to prevent transmission of erroneous characters except to re-start the program.

Once you have a copy of the Baudot listing, take a look at page 3. Addresses 16640 through 16703 contain the TS-1000/Baudot conversion table. The code column contains the TS-1000 print code for a character. The op column shows the corresponding Baudot character sent or received (encoded in the last five bits of the address). The comment column indicates if the character in Baudot is not contained in TS-1000 and what to type as an equivalent. Some of the equivalents are inverse characters. If you include the TR switching option, you will not be able to send these characters because the software necessary for switching will replace the software for getting into and out of "inverse." Inverse is entered and exited by pressing shift-ENTER (FUNCTION). For example, to send the bell signal, type shift-ENTER, shift-B, and shift-ENTER. When the bell signal is received, a normal asterisk is printed.

The automatic sending of the figs and ltrs signals has not been included so you must be sure to type < before any series of numbers or punctuation and > afterwards. To send the text

SO 73 NICK. KL7KA DE KA1VW SK

type

SO <73> NICK <.> KL<7>KA DE KA<1>VW SK.

Additional Features

There are three additional features to point out. The first situation occurs when you are in the receive mode; the figs signal is sometimes received, but for various reasons the following ltrs signal is not received. In this case the system can get hung up printing all sorts of unnecessary numbers and punctuation. Pressing ENTER at any time while in the receive mode sets the system to ltrs. In fact, you can hold the key down and prevent the system from printing any figs. If you do this, any numbers received are printed as their letter equivalent (5 will be T, 7 will be U, ? will be B, and so on).

The carriage return and line feed signals are automatically sent after each set of 63 characters. If you include these signals in the transmitted text, they will **not** force a reset of the counter and will be included in the count. Changing the count requirement or defeating this function is explained in the next section.

As mentioned before, there is a special string, A$, that can be defined explicitly (without a line number). Once defined, it can be loaded into the transmit buffer by pressing shift-A (STOP). If you did not define A$ before saving the program, you may stop the program by typing shift-A. This will cause the error message 2/10, indicating that the string in line 10 of the BASIC program is undefined. If you have already defined A$ and want to change it, type shift-A and then press the space (BREAK) key. **IMPORTANT:** Whenever you stop the program you must type POKE 16595,205 before you do anything else.

Once you've stopped the program and have completed the necessary pokeing, define A$ explicitly. One example is LET A$=" CQ CQ CQ CQ CQ DE KA<1>VW ". A$ may be up to 120 characters long in a 2k RAM system. Type GOTO 2 to re-enter the program. One additional comment, by pressing shift-A you will load A$ into the transmit buffer and change the mode to transmit regardless of what mode you were in. (A special note to users who select the software TR option: Pressing shift-A does everything I have described above, but it will not activate the TR relay. The only way to activate the relay is to press shift-S or shift-D.)

System Operation

Before you type in the machine language code, decide how you want the system to work with re-

spect to the following adjustable conditions. Do you want to use your Timex printer that is limited for use at 45 baud only? If not, you will want to delete the calls to the printer routine that cause irritating screen flicker. This is done by typing in Ø's (NOP - no operation) for the code at the following addresses

16542 16543 16544
16609 16610 16611
16863 16864 16865 16866
16899 16900 16901
17037 17038 17039

As a reminder that you are not going to use the printer, (NOP) will appear at these locations in the comment column.

Do you want to change the automatic carriage return line feed transmission operation? If so, address 17086 contains one more than the number of characters sent between carriage return and line feed signals. If you want to defeat the automatic operation, load code 201 at location 17087. The carriage return or line feed signal is now sent only when included in the typed text. This is very handy for message formatting.

When the line feed signal is received, do you want the display system to ingore it, print a space, or scroll the line? This is controlled by the codes shown at 16843.

The feature that checks for a space after printing 21 characters on the received line may be changed. You may prefer to print a full 32 character line and not care about parts of words at the end of a line. The value that controls when the search begins is located at 16874. If you want a full 32 character line, put 32 into this location.

There is one other area of customizing you may want to experiment with once you've got the system running. There are two timing loops for controlling the rate at which the keyboard is scanned for characters. If you are a fast typist, you may want to lower these values to prevent missing characters. If you are slow at removing your finger from the keys you may need to increase the values to prevent unwanted multiple letters. This means that characters are auto-repeating if you hold the key down. The locations of the time constants are 16578 (receive) and 16712 (transmit). The values included in the listing are for typing at about 45 wpm (comfortable for the author).

The machine language routine is 620 bytes long. It is written to be located starting at address 16514, the first byte in a remark statement that is the first line of a BASIC program. Therefore, you will need to make a REM statement, line one, that has 620 characters in it. I have seen several schemes for making long REM statements but the only reliable method I know of is to type it in directly. This is a long process (in FAST mode), so once you have it, put it on tape. The ASCII program needs 630 bytes so you may want to use it again, and there is always the chance that the system will crash while typing in the code. Once you've got the REM statement, type in the following BASIC program.

```
FAST
1 REM LLL111111111---620 characters
2 FOR N=16514 TO 17131
3 SCROLL
4 PRINT N,
5 INPUT I
6 POKE N,I
7 PRINT PEEK N
8 NEXT N
```

Run this program. As each address appears at the bottom of the display, type in the corresponding code from the Baudot listing. If you make an error, you can stop by pressing shift-A (STOP), explicitly assigning the address of the error to N, then type GOTO 3. The correct code can be re-entered from where the error occured and not from the beginning.

After all of the code has been entered, change the following lines.

```
4 PRINT N, PEEK N
5 NEXT N
6 STOP
SLOW
```

Run it again. As each address and its code appear, check it against the listing. If an error is discovered note the address. After checking, poke any addresses with the correct code as necessary.

Once you are sure the code is correct save what you have temporarily. Then enter the following changes.

```
FAST
2 FOR N=Ø TO 21
3 PRINT AT N,31;" "
4 NEXT N
5 SLOW
6 LET A=USR 16514
7 LET A=USR 16517
8 FAST
9 POKE 16595,201
10 FOR N=1 TO LEN A$
11 POKE 17118, CODE A$(N)
12 LET A=USR 17111
13 NEXT N
14 POKE 16595,205
15 SLOW
16 LET A=USR 16592
17 GOTO 8
18 SAVE "BAUDOT"
19 GOTO 2
```

Turn on your tape recorder. When you're ready to save the program, type GOTO 18. The program saves itself and comes out of the save routine running in receive. If you don't have the serial interface attached, it may crash or get hung in a loop. This is not critical because you've got the program on tape and this should not happen with the interface attached.

If you want to save a particular A$ with the program, type it in without a line number, but before you type GOTO 18. The A$ will be saved on tape with the rest of the program. Remember that

the maximum length of A$ is 120 characters in a 2k byte RAM system.

If you now have more than 2k RAM and you want to have a longer string that's fine. Be sure, however, there is enough room in the buffer to load the whole string thought. I suggest that you keep it under 256 characters. Another way is to define more than one string. You can define 25 strings, B$ through Z$, and call them up by number using the BASIC routine that follows. Use the same lines for lines 2 through 8 as shown above.

```
9 GOTO 12
10 LET A$=B$
11 GOTO 21
12 INPUT Z
13 GOTO Z*10
18 SAVE "BAUDOT"
19 GOTO 2
20 LET A$=C$
21 POKE 16595,201
22 FOR N=1 TO LEN A$
23 POKE 17118, CODE A$(N)
24 LET A=USR 17111
25 NEXT N
26 POKE 16595,205
27 SLOW
28 LET A=USR 16592
29 GOTO 8
30 LET A$=D$
31 GOTO 21
40 LET A$=E$
41 GOTO 21
50 LET A$=F$
51 GOTO 21  ...and so on.
```

This continues for as many strings as you have room for up to Z$. You can define various strings explicitly for RY's, CQ's, brag, ID, or whatever. When you press shift-A an inverse L prompt appears at the bottom of the screen. Enter the number corresponding to the string number that you want to send (1 sends B$, 2 sends C$, 3 sends D$, you get the idea). Be careful that there is enough room in the transmit buffer to hold all of the string.

The ASCII Program

Since the operation of the ASCII software is similar to that of the Baudot software, I will only describe the significant differences. The major difference is that no provision is made to include the use of the printer. The printer control routine written in the BASIC ROM will not operate fast enough to print over 60 WPM. The summary of key functions for the Baudot program apply to the ASCII program as well except that pressing the ENTER key has no effect. This is so because there is no equivalent to figs and ltrs in ASCII. It is tested for in the machine language program and can be used at some time for future features.

The character code conversion table is on pages three and four of the ASCII listing. The Timex character code 136 is used for characters not printed on reception nor capable of being transmitted. These are mostly control characters. If you want or need to include any of these ASCII characters, you will have to swap codes with another character, thereby losing the ability to send that character. Note that graphics characters and keywords codes cannot be used and that lower case ASCII characters are printed as inverse characters.

The characters are sent as seven bits plus parity set to mark and two stop bits. This is compatible with almost all Amateur Radio ASCII communications. No parity checking is done on the received code and only the least significant seven bits are used for character conversion.

The customizing locations are as follows. Automatic carriage return and line feed transmission options are at 17098 and 17099, the line feed interpret options are at 16857, the character count for the scroll on space routine is at 16896, and the time constants for the keyboard scan are at 16566 and 16625.

The machine language routine will require a 630 byte remark statement for line one of the BASIC program. This is ten bytes longer than the Baudot program required. The single string (A$) BASIC program is

```
1 REM 111111—— 630 bytes —— 1111
2 FOR N=Ø TO 21
3 PRINT AT N,31;" "
4 NEXT N
5 SLOW
6 LET A=USR 16514
7 LET A=USR 16517
8 FAST
9 POKE 16583,201
10 POKE 16625,Ø
11 FOR N=1 TO LEN a$
12 POKE 17130, CODE A$(N)
13 LET A=USR 17123
14 NEXT N
15 POKE 16625,3
16 POKE 16583,205
17 SLOW
18 LET A=USR 16580
19 GOTO 8
20 SAVE "ASCII"
21 GOTO 2
```

Very Important: Whenever you stop the program be sure to POKE 16583,205 and POKE 16625,3 before restarting the program.

Summary of Key Functions

ENTER causes received characters to print in ltrs mode.

shift-ENTER causes entry to or exit from printing inverse characters in the transmit buffer.

shift-A causes the predefined string to load into the transmit buffer.

shift-S changes to transmit buffer.

shift-D changes to receive.

To stop program run press shift-A and then press the space (BREAK) key while the string is loading.

References

[1] Listings for both the ASCII and Baudot operation of this program are available for $1 and a business-sized s.a.s.e. Write to QEX, Technical Dept., 225 Main St., Newington, CT 06111.

[2] "Z80 SIO Technical Manual," SGS-ATES Semiconductor Corporation, 240 Bear Hill Rd., Waltham, MA 02154

[3] Carr, Joseph J.,"Z80 Users Manual," Reston Publishing Company, Reston, VA

[4] Zebra Systems, Inc., 78-06 Jamaica Ave., Woodhaven, NY 11421

Code Changes for TR Switching. Note that this eliminates the "inverse" mode.

	ELE	CODE	OP (Comment)
			BAUDOT LISTING
16	538	32	JR NZ 21
	539	21	
	540	62	LD A, 118
	541	118	
	542	205	CALL (2129) (NOP)
	543	81	(NOP)
	544	8	(NOP)
	545	62	LD A, 5
	546	5	
	547	211	OUT (75), A
	548	75	
	549	62	LD A, 136
	550	136	
	551	211	OUT (75), A
	552	75	
	553	24	JR 37
	554	37	
16	604	32	JR NZ 22
	605	22	
	606	62	LD A, 118
	607	118	
	608	205	CALL (2129) (NOP)
	609	81	(NOP)
	610	8	(NOP)
	611	62	LD A, 5
	612	5	
	613	211	OUT (75), A
	614	75	
	615	62	LD A, 8
	616	8	
	617	211	OUT (75), A
	618	75	
	619	24	JR 212
	620	212	
			ASCII LISTING
16	533	32	JR NZ 14
	534	14	
	535	62	LD A, 5
	536	5	
	537	211	OUT (75), A
	538	75	
	539	62	LD A, 232
	540	232	
	541	211	OUT (75), A
	542	75	
	543	24	JR 35
	544	35	
16	592	32	JR NZ 14
	593	14	
	594	62	LD A, 5
	595	5	
	596	211	OUT (75), A
	597	75	
	598	62	LD A, 104
	599	104	
	600	211	OUT (75), A
	601	75	
	602	24	JR 217
	603	217	

September Oscar Program

In the September QZX there was an OSCAR program which in some issues had some lines truncated or blurred. Here are, in better condition, most of those lines.

```
3015 PRINT INK 4;"BREAK TO STOP, CONT OR GOTO 25"
3160 PRINT INK 6;" 9 spaces and a right half shaded graphics block"
5180 RETURN
5260 LET LA=13.75
6730 LET D6=D6+365*(YR-Y2)
```

TABLE 1: Z80A SIO/O PIN FUNCTIONS

PIN	FUNCTION	PIN	FUNCTION
1.	DATA 1	21.	RESET
2.	DATA 3	22.	DATA CARRIER DETECT (B)
3.	DATA 5	23.	CLEAR TO SEND (B)
4.	DATA 7	24.	REQUEST TO SEND (B)
5.	INTERRUPT CPU	25.	DATA TERMINAL READY (B)
6.	INT. ENABLE IN (DAISY CHAIN)	26.	TRANSMIT DATA (B)
7.	INT. ENABLE OUT (DAISY CHAIN)	27.	TRANSMIT & RECEIVE DATA CLOCK (B)
8.	$\overline{M1}$ FROM CPU	28.	RECEIVE DATA (B)
9.	V_{cc} (+5 VDC)	29.	EXT. SYNCH (B)
10.	WAIT/READY (A)	30.	WAIT/READY (B)
11.	EXT. SYNCH (A)	31.	GROUND
12.	RECEIVE DATA (A)	32.	$\overline{RD}$ FROM CPU
13.	RECEIVE DATA CLOCK (A)	33.	COMMAND/DATA SELECT
14.	TRANSMIT DATA CLOCK (A)	34.	CHANNEL SELECT (A/B)
15.	TRANSMIT DATA (A)	35.	CHIP ENABLE
16.	DATA TERMINAL READY (A)	36.	$\overline{IORQ}$ FROM CPU
17.	REQUEST TO SEND (A)	37.	DATA 6
18.	CLEAR TO SEND (A)	38.	DATA 4
19.	DATA CARRIER DETECT (A)	39.	DATA 2
20.	CPU CLOCK	40.	DATA 0

/1 version: 25 is TRANSMIT DATA (B); 26 is TRANSMIT DATA CLOCK (B); 27 is RECEIVE DATA CLOCK (B); DATA TERMINAL READY (B) is not available.

/2 version: 27 is TRANSMIT DATA CLOCK (B); 28 is RECEIVE DATA CLOCK (B); 29 is RECEIVE DATA (B); EXT. SYNCH (B) is not available.

/9 version: single channel only (A); pins 22 through 30 must be left unconnected.

QEX QEX Subscription Order Card

American Radio Relay League

Newington, Connecticut, U.S.A. 06111

For 12 issues of QEX:
In the U.S.
☐ ARRL Member $6.00
☐ Non-Member $12.00
in Canada, Mexico, and U.S. by First Class Mail
☐ ARRL Member $8.40
☐ Non-Member $14.40
Elsewhere by Airmail
☐ ARRL Member $20.00
☐ Non-Member $26.00
Elsewhere by Surface Mail
☐ ARRL Member $9.60
☐ Non-Member $15.60
Remittance must be in U.S. funds and checks must be drawn on a bank in the U.S. Prices subject to change without notice.

QEX, The ARRL Experimenter's Exchange is available at the rates shown at left. Maximum term is 12 issues, and because of the uncertainty of postal rates, prices are subject to change without notice.

☐ Renewal ☐ New Subscription

ARRL Membership Control # _____

Name _____ Call _____

Address _____

City _____ State or Province _____ Zip or Postal Code _____

Profession: _____ Signature _____

Payment Enclosed ☐
Charge to my ☐ Mastercard ☐ VISA ☐ American Express

Account # _____ Valid from _____ Good thru _____ MC Bank # _____

Signature _____ Date _____

QEX1 683

73, W4GHV & K5XY

QZX presents the best software & hardware projects available for the radio amateur. While the contents are believed to be accurate, no warranty is extended on any program or project. No investigation has been made of the claims or reputation of any advertiser. It would be greatly appreciated if all correspondance requiring an answer would be accompanied by a self-addressed stamped envelope (SASE). Copyright by QZX, 1984. All rights reserved, photocopying by permission only.

```
*****************************************
*Please send me QZX for 1 year, starting*
*immediately. Enclosed is $12.00 for    *
*first class mailing. (Those with add-  *
*resses not in Canada, the US, or Mexico*
*send $19.00 in US funds for air mail   *
*delivery.. Thanks  (12/84)             *
*                                       *
* Name _________________________________ *
* Call _____________                     *
*                                       *
* Address ______________________________ *
* .                                     *
* City _________________________________ *
*                                       *
* State _______  ZIP __________          *
*****************************************
```

QZX Journal
2025 O'Donnell Drive
Las Cruces, N.M. 88001

FIRST CLASS POSTAGE

LAS CRUCES, NM
PM
DEC 26
1984